

Санкт-Петербургский государственный университет

**Гусев Егор Витальевич**

**Выпускная квалификационная работа**

*Распознавание наличия медицинской маски на лице человека*

Уровень образования: Бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2020 «Прикладная математика,  
фундаментальная информатика и программирование»

Научный руководитель: Доцент, кафедра ТСУЭФА,

к.ф.-м.н. Головкина Анна Геннадьевна

Рецензент: Профессор, кафедра диагностики

функциональных систем, д.ф. - м.н.

Котина Елена Дмитриевна

Санкт-Петербург

2024 г

## Содержание

<b>Введение</b>	<b>3</b>
<b>Постановка задачи</b>	<b>5</b>
<b>Обзор литературы</b>	<b>6</b>
<b>Глава 1. Обзор датасета и предобработка изображений</b>	<b>7</b>
1.1 Описание датасета	7
1.2 Предобработка изображений	10
<b>Глава 2. Обзор предлагаемых решений</b>	<b>13</b>
2.1 Нейросетевые технологии	13
2.2 Модели детекции	16
2.3 Модели классификации	20
<b>Глава 3. Сравнение предлагаемых решений.</b>	<b>22</b>
3.1 Обучение моделей	22
3.2 Сравнение решений по точности	24
3.3 Сравнение решений по скорости вычисления	29
<b>Выводы</b>	<b>30</b>
<b>Заключение</b>	<b>32</b>
<b>Список использованной литературы</b>	<b>33</b>

## **Введение**

Вопрос безопасности почти постоянно стоит перед всеми людьми, а в современном мире он почти всегда затрагивает компании. Решение этого вопроса предоставляется и совершенствуется с ростом свободы доступа к информации и технологиям, а также уровня вовлеченности людей в эти технологии. Сегодня одним из наиболее востребованных подходов к осуществлению безопасности является использование камер.

Преимущества использования камер очевидны, но для более эффективного использования совмещаются возможности камер записывать видео с возможностями интеллекта человека - распознать, что или кто на этом видео.

Введение в исследование распознавания наличия медицинской маски на лице человека с использованием методов компьютерного зрения обладает особой актуальностью в свете недавних глобальных событий, связанных с пандемией COVID-19. Ношение масок стало не только мерой индивидуальной защиты, но и предметом общественного здравоохранения, что требует контроля и поддержки на многих уровнях.

Компьютерное зрение, как одна из передовых технологий в области искусственного интеллекта, представляет собой мощный инструмент для автоматизации и улучшения точности и скорости множества процессов, включая мониторинг соблюдения правил ношения масок в общественных местах. Эта технология использует алгоритмы машинного обучения и глубокого обучения для анализа изображений или видеоданных с целью определения, надета ли маска на лице человека, и если да, то корректно ли она надета.

Задача распознавания медицинских масок включает в себя несколько этапов: обнаружение и локализация лиц в кадре, определение наличия маски на лице и оценку корректности её расположения. Это вызовы, которые требуют разработки

специализированных моделей компьютерного зрения, способных эффективно функционировать в разнообразных и изменчивых условиях освещения, с различными углами обзора и при разном качестве изображений.

В рамках данной работы будут рассмотрены существующие подходы и алгоритмы, которые применяются для решения этой задачи, а также их преимущества и недостатки. Особое внимание будет уделено анализу того, какие технологии обеспечивают наибольшую точность и скорость в распознавании масок, что крайне важно для реализации эффективных систем мониторинга на практике. Эта работа стремится внести вклад в развитие методов компьютерного зрения, способствуя повышению уровня общественного здоровья и безопасности.

## **Постановка задачи**

В данной работе решается задача распознавания наличия медицинской маски на лице человека с использованием нейронных сетей.

На каждой фотографии присутствуют люди в масках и без масок. Результат работы нейронной сети: определение местоположения лица и распознавание наличия медицинской маски в области лица человека, а если маска присутствует определение правильности ее ношения.

Каждая модель будет оценена на основе точности распознавания, скорости обработки и устойчивости к различным условиям освещения и ракурсам. Окончательный вывод будет включать рекомендации по применению каждой модели в зависимости от конкретных условий использования и требований к системе.

Задачи выпускной квалификационной работы:

1. Сформировать обширный набор данных, который включает изображения лиц с масками и без масок, снятые в различных условиях. Набор данных должен быть размечен соответствующим образом для обучения и тестирования моделей.
2. Провести обзор литературы и выбрать наилучшие решения для сравнения.
3. Обучить каждую из моделей на подготовленном наборе данных.
4. Провести оценку каждой модели на тестовом наборе данных, собрав метрики для оценки точности классификации в области лица. Оценить скорость вычисления моделей.
5. Сравнить производительность всех решений по заданным метрикам для определения наиболее эффективной модели для задачи распознавания масок.

## Обзор литературы

Исследование в области распознавания наличия медицинской маски на лице человека включало в себя обширный анализ существующей литературы.

Для выбора алгоритмов решения данной задачи были проанализированы работы [2-5, 20]. Из данных работ видно, что модель для детекции SSD лучше чем YOLO в точности распознавания объектов, но требует больших вычислительных возможностей. В свою очередь при использовании Mobilenet в качестве основной части в модели SSD мы получаем прирост в производительности, не сильно теряя в качестве распознавания[20].

Модель Faster R-CNN очень требовательная к вычислительным возможностям, но имеет самую высокую точность из рассмотренных в данных работах моделей.

Для выбора модели классификации в качестве основной части для моделей детекции были изучены работы [6-9]. Использование Mobilenetv3 позволяет модели детекции понизить требовательность к высоким вычислительным возможностям с незначительным понижением точности. В свою очередь использование более мощной модели классификации VGG16 в модели детекции SSD может повысить точность при возможности распознавания в режиме реального времени.

Для реализации данных алгоритмов и оценки качества распознавания моделей была рассмотрена литература [10-16]. Это включало знакомство с библиотеками и основными возможностями языка программирования Python.

Исследования проводились на наборе данных, взятого из открытого источника [1], состоящем из фотографий людей в масках и без масок, на улицах и в помещениях с различными условиями освещения и в разных позах.

# Глава 1. Обзор датасета и предобработка изображений

## 1.1 Описание датасета

Датасет для обнаружения масок на лице включает разнообразные изображения, которые охватывают различные условия и контексты, чтобы имитировать реалистичные сценарии.

### Позы людей.

Изображения включают лица людей в различных позах. Это может включать лица, смотрящие прямо на камеру, а также повернутые в разные стороны лица (профильные и полупрофильные) (Рис. 1). Такое разнообразие поз помогает создать более универсальный датасет для обучения алгоритмов, способных распознавать наличие масок в различных условиях.



Рис.1. Пример данных для фотографии людей в различных позах.

### Цвета масок и их типы.

Маски в датасете могут варьироваться по цвету и типу (Рис. 2):

1. Цвета: Чёрные, белые, синие и маски разнообразных цветов и с разными узорами. Это помогает обучить модель распознавать маски независимо от их цвета.
2. Типы масок: Включая одноразовые медицинские маски, тканевые маски и даже респираторы с клапанами.



Рис. 2. Пример данных для фотографий людей в различных масках.

### **Классы.**

Лица в датасете обычно разделены на несколько классов:

1. С маской: лицо полностью закрыто маской в соответствии с рекомендациями по ношению.
2. Без маски: лицо не закрыто маской.
3. Маска надета неправильно: маска надета, но не полностью закрывает необходимые области лица, например, нос остаётся открытым.

### **Освещение.**

Датасет включает изображения, сделанные при различных условиях освещения (Рис. 3):

1. При естественном свете и в помещениях с искусственным освещением.
2. Изображения, где лица освещены равномерно, а также экземпляры с тенями или противосветом.

### **Фон.**

Фон на изображениях может быть как однородным, так и сложным, что включает (Рис. 3):

1. Уличные сцены, интерьеры помещений.
2. Бездействующий и динамичный бэкграунд, позволяя обучающим моделям выделять лица на любом фоне.

Датасет с таким разнообразием условий отлично подходит для разработки и тестирования систем обнаружения и распознавания масок на лице в различных реальных условиях использования.



Рис. 3. Пример данных для фотографий людей в помещении / на улице с разной степенью освещенности.

## 1.2 Предобработка изображений

Предобработка изображений - это этап обработки изображений, который выполняется перед применением алгоритмов распознавания, классификации, детекции или других методов анализа изображений. Этот процесс включает в себя различные операции, направленные на улучшение качества и уменьшение шума в изображении, а также на выделение важных признаков, которые могут быть использованы в последующих этапах анализа.

Из-за того, что фотографии могут быть сделаны в условиях разной освещенности, необходимо провести предобработку изображения перед непосредственным попаданием изображения в модель для распознавания наличия медицинской маски. Для этого был использован алгоритм CLAHE[16] для выравнивания контрастности изображения.

CLAHE (Contrast Limited Adaptive Histogram Equalization) - это метод повышения контраста изображения, который улучшает традиционные методы гистограммного выравнивания (HE) и адаптивного гистограммного выравнивания (AHE). CLAHE ограничивает усиление контраста для предотвращения усиления шума.

### Шаги алгоритма CLAHE

#### 1. Разделение изображения:

Входное изображение делится на более мелкие, не перекрывающиеся регионы, называемые плитками или блоками. Обычно каждая плитка имеет размер  $M \times N$ .

#### 2. Вычисление гистограммы:

Для каждой плитки вычисляется гистограмма интенсивностей пикселей. Гистограмма  $H(i)$  для уровня интенсивности  $i$  в плитке  $t$  задается формулой:

$$H_t(i) = \sum_{(x,y) \in \text{плитка } t} \delta(I(x,y) - i),$$

где  $I(x, y)$  - интенсивность пикселя в координатах  $(x, y)$ , а  $\delta$  - дельта-функция Кронекера, равная 1, если аргумент равен 0, и 0 в противном случае.

3. Ограничение гистограммы:

Ограничивается высота гистограммы, чтобы избежать усиления шума. Это делается путем задания предельного значения контраста  $C$ . Если количество пикселей в любом бине гистограммы превышает  $C$ , то этот бин "обрезается", а излишек равномерно распределяется по другим бинам.

4. Вычисление кумулятивной распределенной функции ( $CDF$ ):

Для каждой плитки вычисляется нормализованная кумулятивная распределенная функция, которая используется для получения функции преобразования для выравнивания гистограммы

$$CDF_t(i) = \frac{\sum_{j=0}^i H_t(j)}{\sum_{j=0}^{L-1} H_t(j)},$$

где  $L$  - количество уровней интенсивности (например, 256 для 8-битного изображения).

5. Преобразование пикселей:

Интенсивности пикселей в каждой плитке преобразуются с использованием  $CDF$ , чтобы получить новые значения интенсивности

$$I_{new}(x, y) = CDF_t(I(x, y)) \cdot (L - 1).$$

6. Интерполяция краев:

Для предотвращения артефактов на границах плиток используется билинейная интерполяция. Интенсивности пикселей, попадающих на границы плиток, интерполируются между соседними плитками.

Пример работы алгоритма CLANE (Рис. 4):

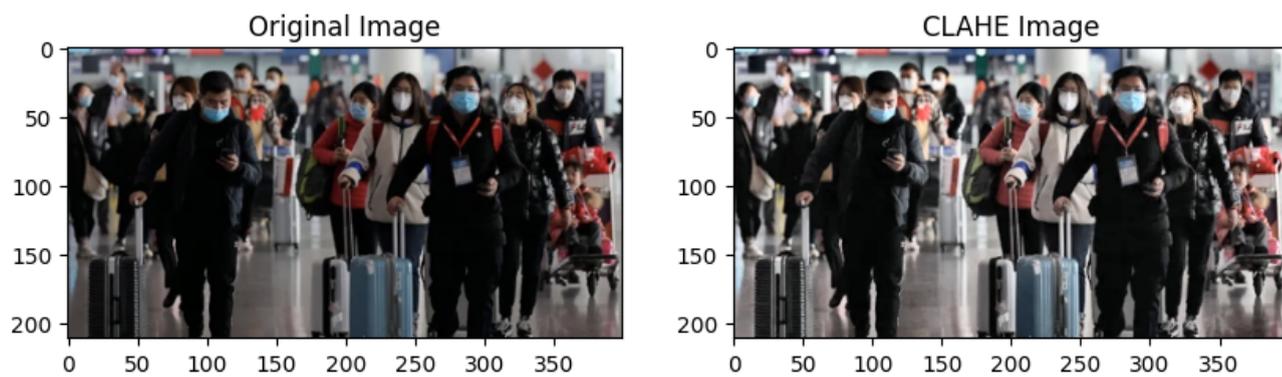


Рис. 4. Пример работы алгоритма CLAHE для выравнивания контрастности изображения.

## Глава 2. Обзор предлагаемых решений

### 2.1 Нейросетевые технологии

Глубокое обучение[10] — это функция искусственного интеллекта, которая имитирует работу человеческого мозга при обработке данных, при обнаружении объектов, распознавании речи, переводе языков и принятии решений.

Нейронные сети[10] используют иной подход к решению задач, чем обычные компьютеры. Обычные компьютеры используют алгоритмический подход, то есть компьютер следует набору инструкций для того, чтобы решить проблему. Если не известны конкретные шаги, которые необходимо выполнить компьютеру, компьютер не сможет решить проблему. Это ограничивает возможности обычных компьютеров по решению проблем проблемами, которые мы уже понимаем и знаем, как их решать. Но компьютеры были бы намного полезнее, если бы они могли делать то, чего мы точно не знаем, как делать. Нейронные сети обрабатывают информацию аналогично тому, как это делает человеческий мозг.

Сеть состоит из большого числа тесно взаимосвязанных обрабатывающих элементов (нейронов), работающих параллельно для решения конкретной задачи. Нейронные сети учатся на примере[11]. Они не могут быть запрограммированы на то, чтобы тратиться впустую или, что еще хуже, сеть могла бы функционировать неправильно. Недостатком является то, что, поскольку сеть сама находит способ решить проблему, ее работа может быть непредсказуемой. С другой стороны, обычные компьютеры используют когнитивный подход к решению проблемы; способ решения проблемы должен быть известен и изложен в небольших недвусмысленных инструкциях. Затем эти инструкции преобразуются в программу на языке высокого уровня, а затем в машинный код, понятный компьютеру.

Эти машины полностью предсказуемы; если что-то пойдет не так, это произойдет из-за программного или аппаратного сбоя. Нейронные сети и обычные алгоритмические компьютеры не конкурируют, а дополняют друг друга. Есть задачи, которые больше подходят для алгоритмического подхода, такие как арифметические

операции, и задачи, которые больше подходят для нейронные сети. Более того, для решения большого числа задач требуются системы, использующие комбинацию двух подходов (обычно для управления нейронной сетью используется обычный компьютер), чтобы работать с максимальной эффективностью.

Сверточная нейронная сеть[10] – класс нейронных сетей, который специализируется на обработке данных, имеющих топологию в виде сетки, например изображений. Цифровое изображение – это двоичное представление визуальных данных. Он содержит серию пикселей, расположенных в виде сетки, где каждая ячейка содержит визуальные данные: яркость и цвет.

Человеческий мозг обрабатывает огромное количество информации при просмотре изображения. Каждый нейрон работает в своем собственном рецептивном поле и связан с другими нейронами таким образом, что они покрывают все поле зрения. Подобно тому, как каждый нейрон реагирует на стимулы только в ограниченной области поля зрения, называемой рецептивным полем в системе биологического зрения, каждый нейрон в CNN также обрабатывает данные только в своем рецептивном поле. Слои расположены таким образом, что сначала они обнаруживают более простые узоры, а затем более сложные узоры.

Архитектура CNN[10] обычно состоит из нескольких слоев, таких как сверточные слои, слои пулинга и полносвязные слои.

Сверточные слои извлекают пространственные признаки из изображений с помощью фильтров или ядер свертки. Эти фильтры обучаются на основе данных и автоматически выделяют важные признаки, связанные с медицинской маской.

Следующие после сверточных слои пулинга уменьшают размер признаковых карт, сохраняя важные информационные признаки. Это помогает уменьшить количество параметров и вычислительную сложность модели.

Полносвязные слои принимают выходные данные из сверточных и пулинг слоев и связывают их с классификационными слоями. Они преобразуют

высокоуровневые признаки в вероятности наличия или отсутствия медицинской маски на изображении.

Модель CNN обучается на основе размеченного набора данных[10]. Она оптимизируется с использованием алгоритмов оптимизации, таких как стохастические.

## 2.2 Модели детекции

### Модель Faster R-CNN

Одной из нейросетей, используемой в данной работе, является Faster R-CNN[19]. Она состоит из двух модулей. Первый модуль представляет собой глубокую полностью конволюционную сеть, которая предлагает регионы, а второй модуль - детектор, который использует предложенные регионы. Вся система представляет собой единую, унифицированную сеть для обнаружения объектов (Рис. 5).

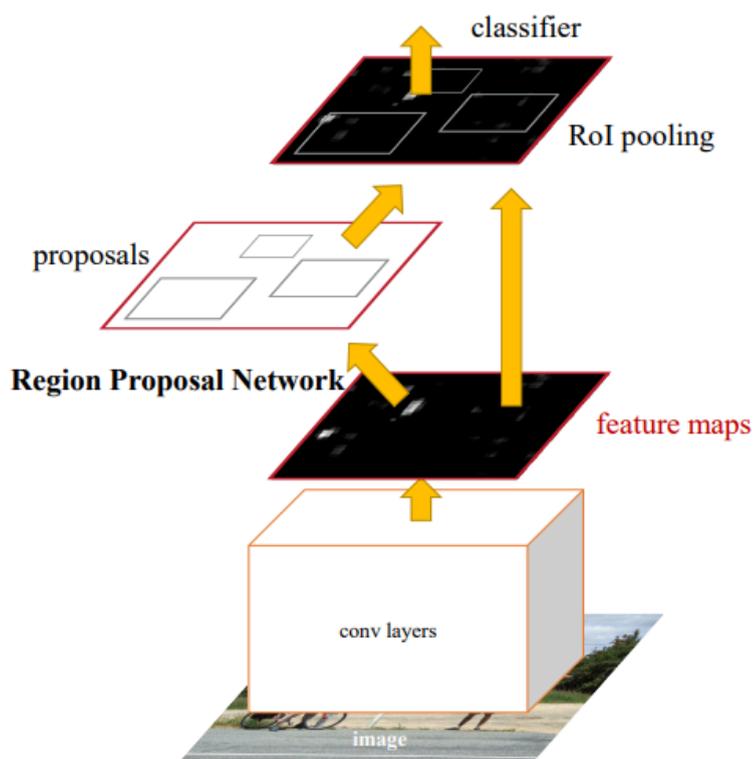


Рис. 5. Общая схема работы модели детекции Faster R-CNN.

Сеть предложения областей (RPN) принимает изображение (любого размера) в качестве входного сигнала и выдает набор прямоугольных объектов, каждое из которых имеет оценку объектности.

Для генерации предложений по регионам создается небольшая сеть по конволюционной карте признаков. Эта небольшая сеть принимает на вход  $n \times n$  пространственное окно входной конволюционной карты признаков. Каждое скользящее окно сопоставляется с низкоразмерным признаком. Эти признаки поступают в два родственных полносвязных слоя - слой бокс-регрессии (*reg*) и слой бокс-классификации (*cls*)(Рис. 6).

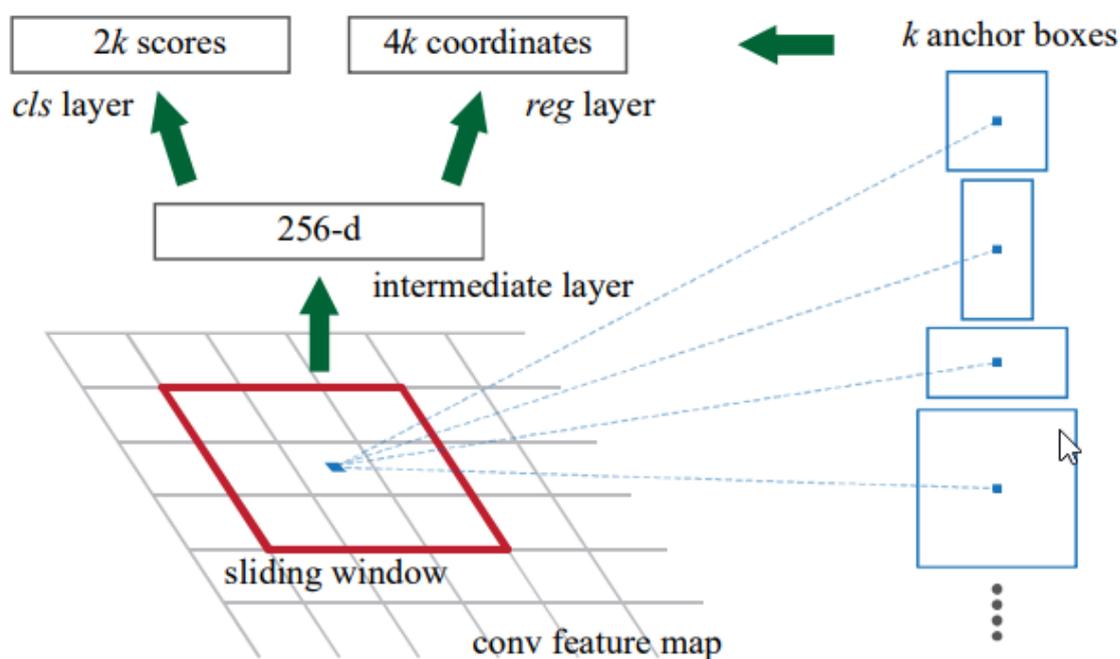


Рис. 6. Пример генерации предложений по регионам в модели Faster R-CNN.

После того как RPN отфильтровывает и ранжирует предложения о регионах, следующим шагом является применение операции RoI Pooling. Этот процесс заключается в извлечении и стандартизации признаков из каждого предложенного региона до фиксированного размера, который можно обработать с помощью полносвязных слоев. RoI Pooling гарантирует, что признаки из регионов различного размера и соотношения сторон преобразуются в однородный вид.

После слоя ROI Pooling последуют полносвязные слои, которые выполняют две ключевые функции:

- Классификация: Определение класса объекта в каждом ROI.
- Регрессия ограничивающей рамки: Коррекция и уточнение предложенных регионов для точного обрамления объектов.

### **Модель SSD**

Подход SSD[18] основан на конволюционной сети с обратной связью, которая генерирует фиксированный размер коллекции ограничительных ячеек и оценок за присутствие экземпляров класса объектов в этих ячейках, после чего следует шаг подавления без максимума для получения окончательных обнаружений. Первые слои сети основаны на стандартной архитектуре, используемой для высококачественной классификации изображений, которая называется базовой сетью (например, VGG16). После основных слоёв добавляются дополнительные сверточные слои, которые уменьшают размерность и увеличивают глубину карт признаков. Эти слои помогают обнаруживать объекты на разных масштабах. После дополнительных сверточных слоев идут предсказательные слои, они используют карты признаков для предсказания классов и ограничивающих рамок для различных якорных рамок, ассоциированных с каждым местоположением на карте.

Основной особенностью SSD является то, что SSD использует якорные рамки на каждом уровне карты признаков для предсказания ограничивающих рамок и классов, что позволяет лучше обрабатывать объекты различных масштабов.

Более полное описание архитектуры SSD представлено на рисунке 7.

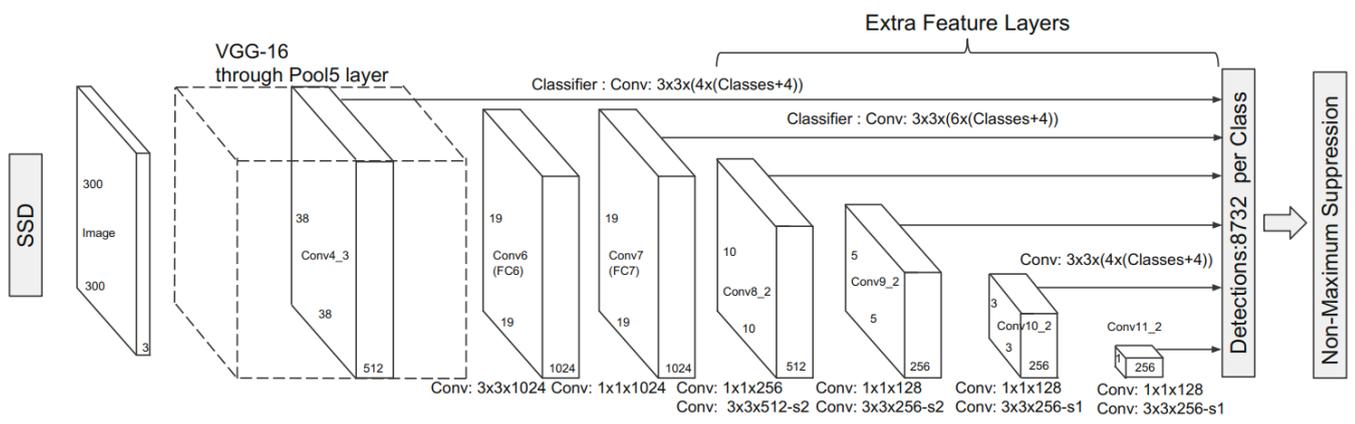


Рис. 7. Архитектура модели детекции SSD.

## 2.3 Модели классификации

### Модель VGG16

VGG16[17] состоит из 16 слоев, включая 13 сверточных слоев и 3 полностью связанных слоя.

Каждый сверточный слой в VGG16 использует фильтры размером  $3 \times 3$  с шагом 1, что позволяет сохранять пространственный размер признаков через слои. Это стандартное применение фильтров делает архитектуру простой и эффективной. После каждого блока сверточных слоев следует слой Max Pooling с размером фильтра  $2 \times 2$  и шагом 2, который уменьшает размерность признаков вдвое. Это сокращение размерности помогает уменьшить количество параметров и вычислительную сложность в глубоких слоях сети.

На конечном этапе сети расположены три полносвязных слоя. Первые два полносвязных слоя имеют по 4096 нейронов каждый и функции активации ReLU, а последний полносвязный слой, который используется для классификации, содержит 1000 нейронов.

В полносвязных слоях используется метод Dropout для предотвращения переобучения. Это важно, учитывая глубину сети и большое количество параметров.

Более полное описание архитектуры VGG16 представлено на рисунке 8.

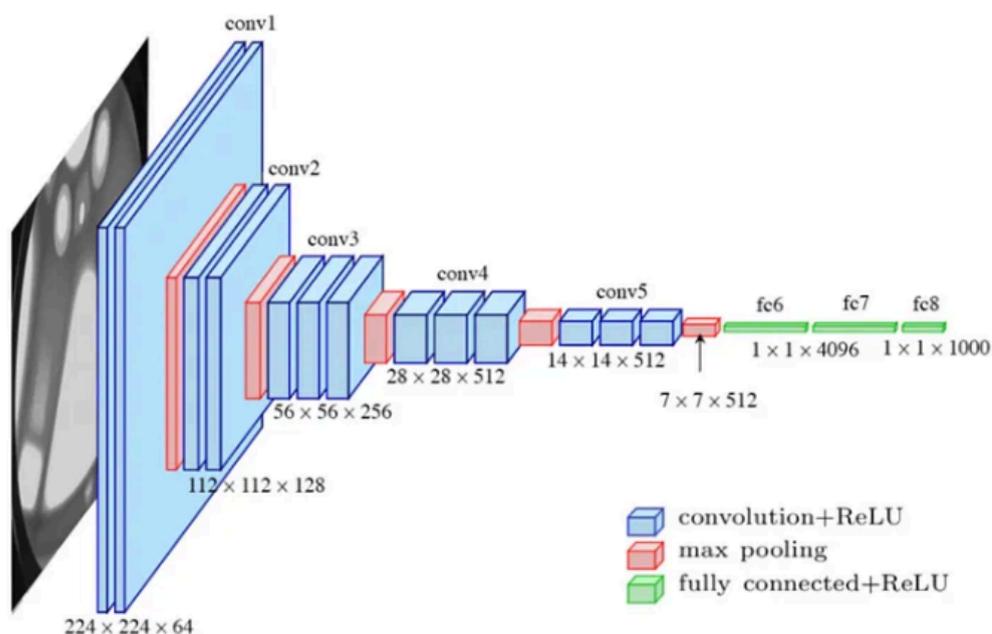


Рис. 8. Архитектура модели классификации VGG16.

### Модель Mobilenetv3

MobileNetV3 — это архитектура нейронной сети, оптимизированная для использования в мобильных и ресурсозависимых устройствах. Ее разработка включала автоматизированный поиск архитектуры и ручные улучшения для достижения оптимального баланса между точностью и производительностью.

Алгоритм NetAdapt используется для точной настройки числа фильтров на каждом уровне сети. Этот метод предлагает и оценивает модификации сети, выбирая лучшие на основе их влияния на задержку и точность.

Начальные и конечные слои оптимизированы для снижения вычислительных затрат. Например, конечный 1x1 сверточный слой перемещен после слоя среднего пула, чтобы уменьшить пространственное разрешение и снизить вычислительную нагрузку без потери точности.

Введена модифицированная функция активации swish, названная hard-swish (h-swish) (Рис. 9). Она быстрее вычисляется и более подходит для квантования, что важно для мобильных устройств.

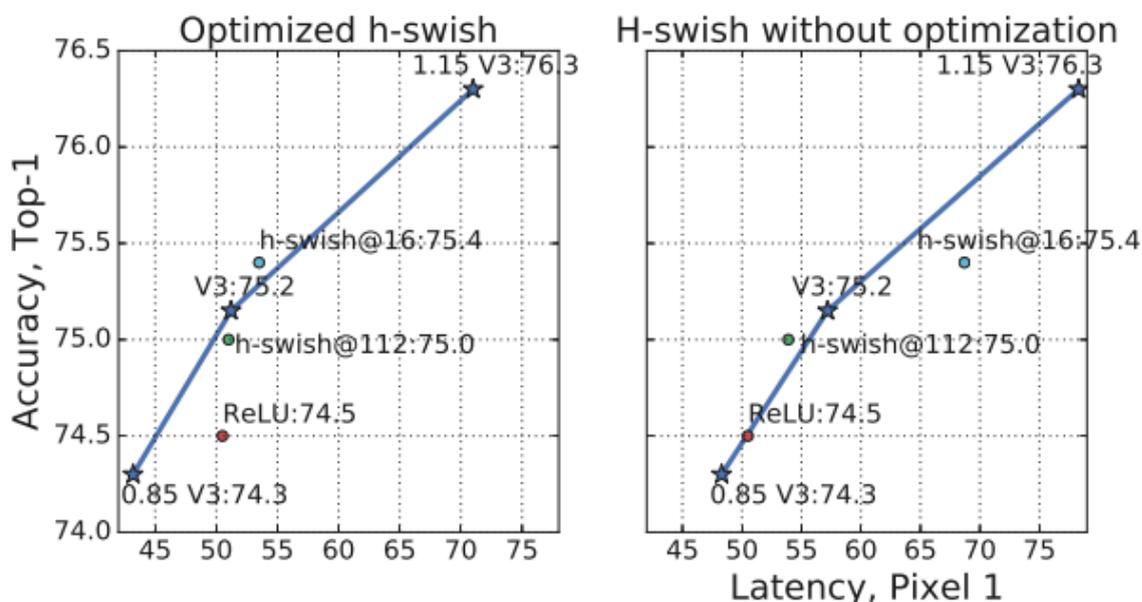


Рис. 9. Влияние h-swish и ReLU на задержку для оптимизированного и неоптимизированный h-swish.

## Глава 3. Сравнение предлагаемых решений.

### 3.1 Обучение моделей

Обучение моделей SSD и Faster R-CNN с использованием оптимизатора SGD (стохастический градиентный спуск) является многоэтапным процессом, который включает в себя подготовку данных, определение моделей, выбор функций потерь, итеративное обновление параметров модели и оценку результатов. Оптимизатор SGD является важным компонентом этого процесса, обеспечивая эффективное обновление параметров модели для достижения лучших результатов на задаче обнаружения объектов.

Для обучения Faster R-CNN и SSD использовалась комбинация функций потерь, которая включает в себя две основные составляющие:

1. Функция потерь для классификации объектов (Classification Loss):

Эта функция потерь оценивает разницу между предсказанными классами объектов и их истинными метками. Обычно для задачи обнаружения объектов применяется функция потерь, такая как кросс-энтропия (cross-entropy loss), которая измеряет различие между распределением вероятностей предсказанных классов и распределением вероятностей истинных классов.

2. Функция потерь для регрессии ограничивающих рамок (Bounding Box Regression Loss):

Эта функция потерь оценивает разницу между предсказанными координатами ограничивающих рамок (bounding boxes) и их истинными координатами. Обычно используется функция потерь, например, среднеквадратичная ошибка (mean squared error), которая измеряет различие между предсказанными и истинными координатами ограничивающих рамок.

Общая функция потерь представляет собой сумму этих двух компонент:

$$Total Loss = Classification Loss + \lambda \cdot Bounding Box Regression Loss,$$

где  $\lambda$  - это коэффициент, который используется для балансировки вклада каждой компоненты в общую функцию потерь.

На рисунке 10 представлен график зависимости функции потерь и точности (accuracy) от пройденных эпох:

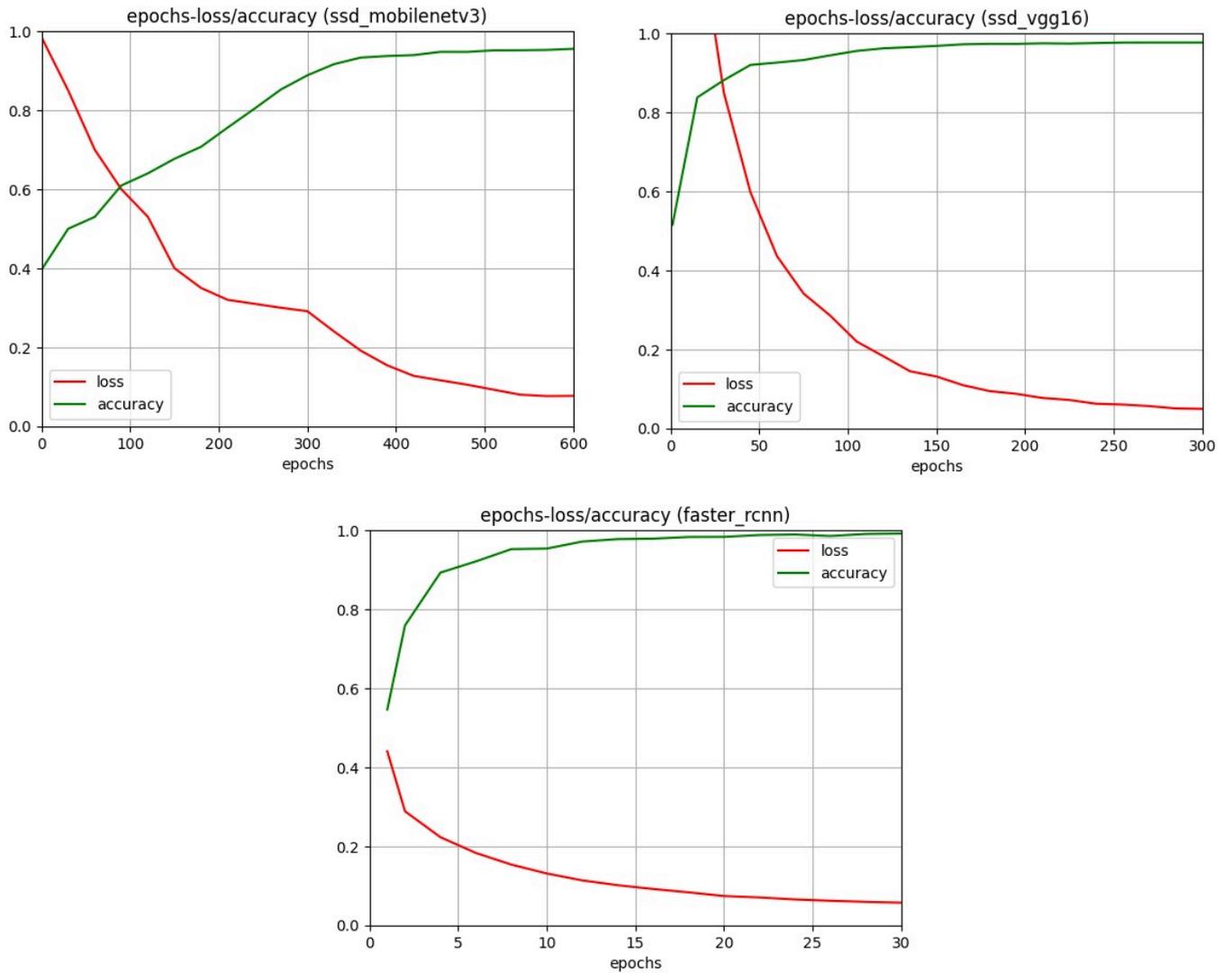


Рис. 10. График зависимости функции потерь и точности (accuracy) от пройденных эпох.

## 3.2 Сравнение решений по точности

Оценка эффективности моделей производится с помощью подсчета метрики f1-score, а также построения F1-confidence curve.

F1-score — это гармоническое среднее точности (precision) и полноты (recall), которое помогает уравновесить вклад обоих этих показателей.

$$F1 - score = 2 \cdot \frac{precision \cdot recall}{precision + recall},$$

где precision - точность, recall - полнота.

$$precision = \frac{TP}{TP + FP},$$

где TP - количество истинно положительных результатов (модель правильно предсказала положительный класс), FP - количество ложно положительных результатов (модель ошибочно предсказала положительный класс).

$$recall = \frac{TP}{TP + FN},$$

где FN - количество ложно отрицательных результатов (модель ошибочно предсказала отрицательный класс, когда он должен быть положительным)

Precision показывает, какая доля предсказанных положительных результатов является истинно положительными.

Recall показывает, какая доля истинно положительных объектов была правильно предсказана моделью.

F1-score особенно полезен, когда распределение классов неравномерно, поскольку он не учитывает истинно отрицательные результаты (true negatives), которые могут быть в большом количестве в доминирующем классе и исказить

общую оценку точности. А в исходном датасете у нас следующее распределение по классам (Рис. 11):

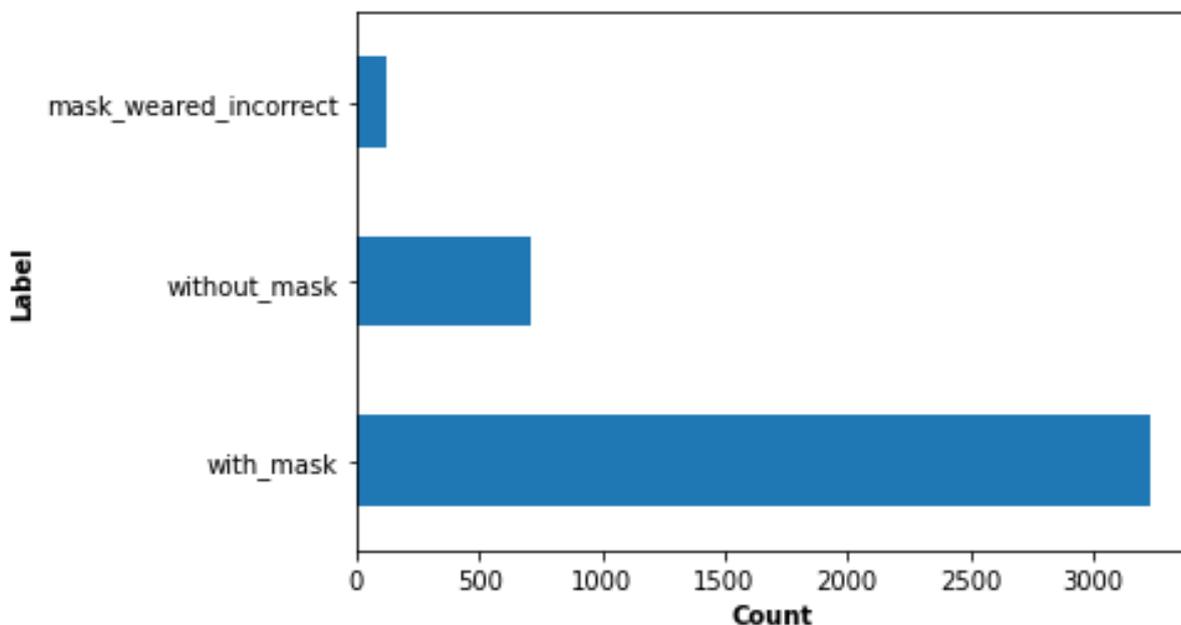


Рис. 11. Распределение классов в датасете.

Градация F1-score для оценки точности модели:

1.  $0.9 < \text{F1-score} \leq 1$  - отличное значение F1-score, модель демонстрирует высокую точность и полноту, ошибки минимальны
2.  $0.8 < \text{F1-score} \leq 0.9$  - хорошее значение F1-score, модель работает хорошо, с допустимыми ошибками
3.  $0.7 \leq \text{F1-score} \leq 0.8$  - приемлемое значение F1-score, модель демонстрирует средние результаты, указывая на необходимость дальнейшей доработки и оптимизации
4.  $\text{F1-score} < 0.7$  - модель имеет низкую точность, требующую значительных улучшений для практического применения.

F1-confidence curve может быть интерпретирована как визуализация, показывающая, как F1-score изменяется в зависимости от порога уверенности

(confidence threshold), используемого для классификации объектов. В этом контексте "уверенность" относится к вероятности, с которой модель присваивает класс определенному примеру. Изменяя порог уверенности, при котором принимаются решения о классификации, можно увидеть, как изменяется баланс между точностью и полнотой, а следовательно, и F1-score. F1-confidence curve может быть полезной для оценки производительности моделей классификации, особенно в задачах, где имеется дисбаланс классов или когда важно найти оптимальный баланс между точностью (precision) и полнотой (recall).

F1-confidence curve для разных классов для SSD+Mobilenetv3, SSD+VGG16, Faster R-CNN (Рис. 12):

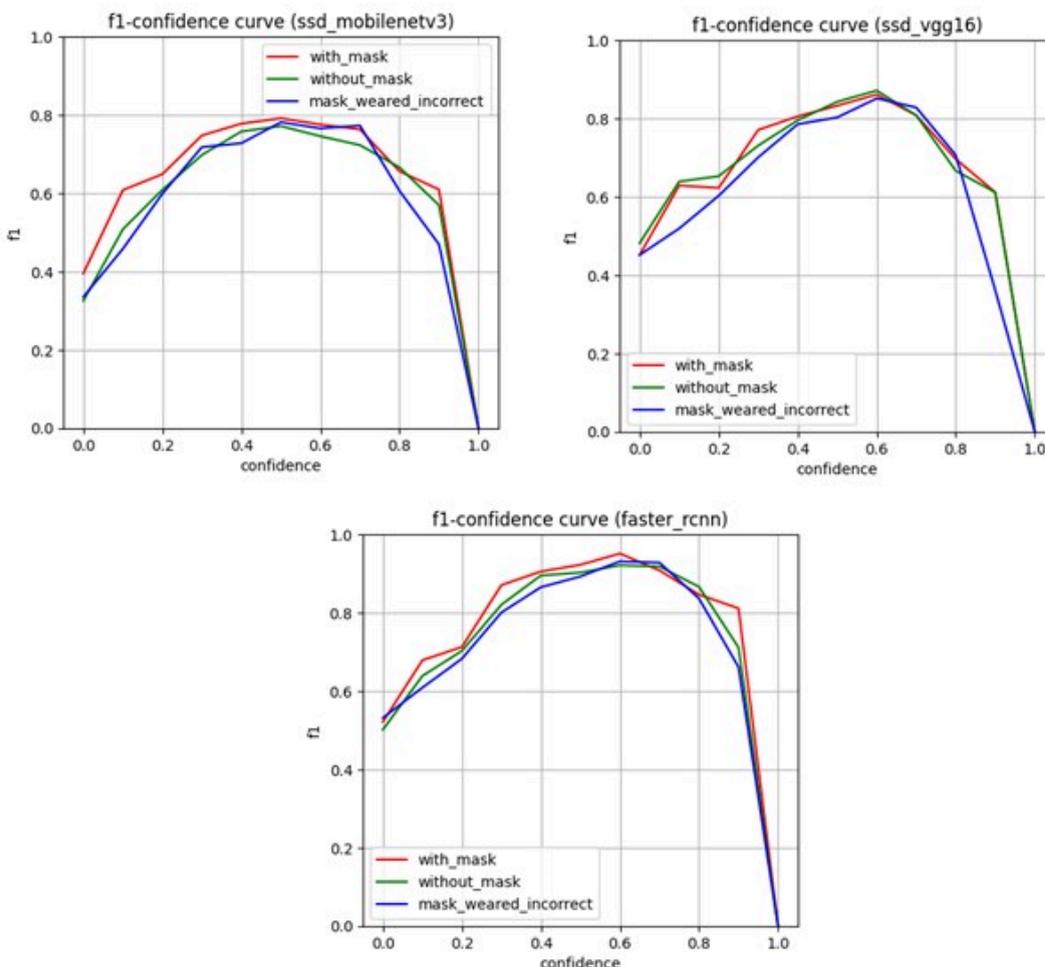


Рис. 12. F1-confidence curve для разных классов для SSD+Mobilenetv3, SSD+VGG16, Faster R-CNN.

Модель SSD+Mobilenetv3 имеет приемлемое, близкое к хорошему значению F1-score для каждого класса, а именно при мере уверенности (confidence) “0.5” - “0.795” для класса “маска надета”, “0.789” для класса “маска не надета”, “0.782” “маска надета неправильно”.

Модель SSD+VGG16 имеет хорошее значение F1-score для каждого класса, а именно при мере уверенности (confidence) “0.6” - “0.845” для класса “маска надета”, “0.852” для класса “маска не надета”, “0.83” “маска надета неправильно”.

Модель Faster R-CNN имеет отличное значени F1-score для каждого класса, а именно при мере уверенности (confidence) “0.6” - “0.93” для класса “маска надета”, “0.916” для класса “маска не надета”, “0.92” “маска надета неправильно”.

Для точного сравнения моделей по точности F1-score для всех классов был подсчитан как среднее значение F1-score для этих классов. Аналогичным способом была построена F1-confidence curve для всех классов для SSD+Mobilenetv3, SSD+VGG16, Faster R-CNN (Рис. 13):

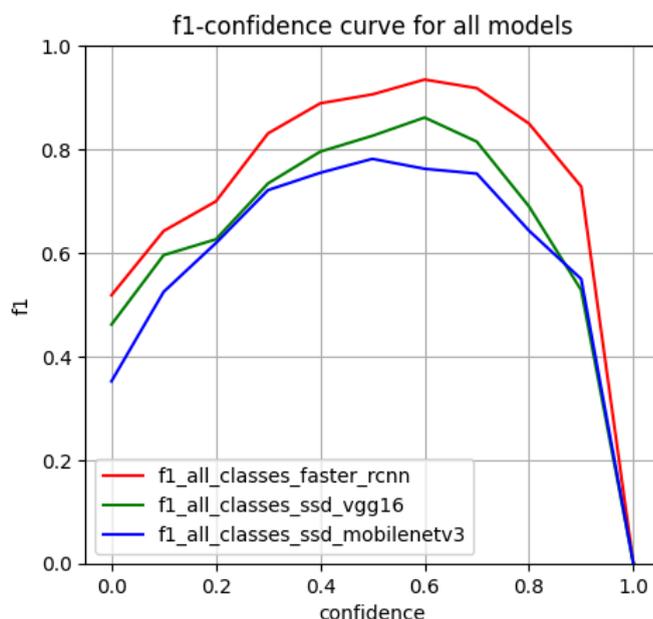


Рис. 13. F1-confidence curve для всех классов для SSD+Mobilenetv3, SSD+VGG16, Faster R-CNN.

Теперь выбрав меру уверенности (confidence) так, чтобы F1-score достигал своего максимального значения можно составить следующую таблицу (Табл. 1):

Модель	Мера уверенности	F1
SSD+Mobilenetv3	0.5	0.79
SSD+VGG16	0.6	0.85
Faster R-CNN	0.6	0.92

Табл. 1. Максимальное значение F1-score для всех классов для моделей SSD+Mobilenetv3, SSD+VGG16, Faster R-CNN.

### 3.3 Сравнение решений по скорости вычисления

Сравнение моделей SSD+Mobilenetv3, SSD+VGG16, Faster R-CNN по скорости вычислений является важным аспектом при выборе подходящего решения для конкретного применения. Скорость инференции, выражаемая в кадрах в секунду, критична для приложений, работающих в режиме реального времени

Сравнение по скорости вычисления проводилось на 200 фотографиях. Результаты для моделей представлены в таблице 2:

Модель	Скорость вычисления (кадры в секунду)
SSD+Mobilenetv3	56
SSD+VGG16	12
Faster R-CNN	6

Табл. 2. Скорость вычисления для моделей SSD+Mobilenetv3, SSD+VGG16, Faster R-CNN.

## Выводы

В данной работе была рассмотрена задача распознавания наличия медицинской маски на лице человека. Для этого было представлено решение с помощью различных моделей, а именно SSD+Mobilenetv3, SSD+VGG16 и Faster R-CNN и проведено сравнение по точности и скорости вычислений. Для этого была подсчитана метрика F1-score и скорость вычисления в кадрах в секунду.

SSD+Mobilenetv3 менее точная чем SSD+VGG16 на 7.5% и менее точная чем Faster R-CNN на 16.5%, но в свою очередь быстрее чем SSD+VGG16 в 4.6 раз и быстрее чем Faster R-CNN в 9.3 раза.

SSD+VGG16 точнее чем SSD+Mobilenetv3 на 7.5% и менее точная чем Faster R-CNN на 8.2%. По скорости вычисления медленнее в 4.6 раза чем SSD+Mobilenetv3 и быстрее в 2 раза чем Faster R-CNN.

Faster R-CNN точнее чем SSD+Mobilenetv3 на 16.5% и на 8.2% - чем SSD+VGG16, но медленнее в 9.3 и 2 раза соответственно.

Причины таких отличий в том, что:

1. MobileNetV3 — это легкая и эффективная модель, которая оптимизирована для мобильных и встроенных приложений, что позволяет ей работать очень быстро при сохранении приемлемой точности.
2. VGG16 — это более крупная и тяжелая модель по сравнению с MobileNetV3. Она требует больше вычислительных ресурсов, что снижает скорость инференции.
3. Faster R-CNN является двухступенчатой моделью детекции объектов, которая сначала генерирует кандидаты регионов (Region Proposals), а затем классифицирует их. Этот процесс более вычислительно затратный по сравнению с SSD.

Для систем со слабыми вычислительными возможностями следует использовать модель SSD+Mobilenetv3. Данная модель может использоваться для систем мониторинга соблюдения правил ношения масок в условиях пандемии, она

имеет приемлемое, близкое к хорошему значению метрики F1-score при незатратных вычислениях.

Для систем со средними-высокими вычислительными возможностями лучше использовать модель SSD+VGG16. Она обладает более высокой точностью детекции и классификации с более затратными вычислениями, но достаточными для работы в реальном времени.

В свою очередь модель Faster R-CNN показала отличный результат по метрике F1-score и стала самой точной из представленных моделей, но имеет очень затратные вычислительные возможности. Из этого следует, что для решения задачи распознавания наличия медицинской маски на лице человека в условиях реального времени она не подходит, но будет предпочтительнее других моделей в статистических и научных вычислительных задачах.

## **Заключение**

В рамках данной работы были обучены на предоставленном датасете, сравнены между собой по точности и времени вычисления модели детекции SSD+Mobilenetv3, SSD+VGG16, Faster R-CNN.

В ходе работы были получены следующие результаты:

1. Сравнение по точности было представлено с помощью подсчета метрики F1-score.
2. Сравнение по времени вычисления было представлено с помощью подсчета количества кадров в секунду, которое может обработать модель.
3. Таким образом были проанализированы результаты и даны рекомендации по применению той или иной рассмотренной модели для разных систем.

Полученные результаты имеют возможность применения моделей для тех или иных целей в зависимости от имеющихся вычислительных возможностей, описанных в предыдущей главе, в области здравоохранения.

## Список использованной литературы

1. Dataset // Kaggle URL: <https://www.kaggle.com/datasets/andrewmvd/face-mask-detection>.
2. Mohammed Ali F. A., Al-Tamimi M. S. H. Face mask detection methods and techniques: A review //International Journal of Nonlinear Analysis and Applications. – 2022. – Т. 13. – №. 1. – С. 3811-3823.
3. Rahman M. H. et al. Real-time face mask position recognition system based on MobileNet model //Smart health. – 2023. – Т. 28. – С. 100382.
4. Said Y. Pynq-YOLO-Net: An embedded quantized convolutional neural network for face mask detection in COVID-19 pandemic era //International Journal of Advanced Computer Science and Applications. – 2020. – Т. 11. – №. 9.
5. Cheng L. et al. Improved SSD network for fast concealed object detection and recognition in passive terahertz security images //Scientific Reports. – 2022. – Т. 12. – №. 1. – С. 12082.
6. Su X. et al. Face mask detection and classification via deep transfer learning //Multimedia Tools and Applications. – 2022. – С. 1-20.
7. Shukla R. K., Tiwari A. K. Masked face recognition using mobilenet v2 with transfer learning //Computer Systems Science & Engineering. – 2023. – Т. 45. – №. 1.
8. Rahman M. H. et al. Real-time face mask position recognition system based on MobileNet model //Smart health. – 2023. – Т. 28. – С. 100382.
9. Zhao L., Wang L. A new lightweight network based on MobileNetV3 //KSII Transactions on Internet and Information Systems (TIIS). – 2022. – Т. 16. – №. 1. – С. 1-15.

10. Джулли А, Пал С. Библиотека Keras - инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow/ пер. с англ. М.: ДМК Пресс, 2018. - 294 с.
11. Рашка С. Python и машинное обучение / пер. с англ. М.: ДМК Пресс, 2017. - 418 с.
12. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow / пер. с англ. СПб.: ООО Альфа-книга, 2018. - 688 с.
13. Python // Python official page. — Access mode: <https://www.python.org/>
14. OpenCV // OpenCV official page – Accessed mode: <https://opencv.org/>
15. PyTorch // PyTorch official page – Access mode: <https://pytorch.org/>
16. Reza A. M. Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement //Journal of VLSI signal processing systems for signal, image and video technology. – 2004. – Т. 38. – С. 35-44.
17. Karen S. Very deep convolutional networks for large-scale image recognition //arXiv preprint arXiv: 1409.1556. – 2014.
18. Liu W. et al. Ssd: Single shot multibox detector //Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. – Springer International Publishing, 2016. – С. 21-37.
19. Ren S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks //Advances in neural information processing systems. – 2015. – Т. 28.
20. Chiu Y. C. et al. Mobilenet-SSDv2: An improved object detection model for embedded systems //2020 International conference on system science and engineering (ICSSE). – IEEE, 2020. – С. 1-5.